Perturbation-Minimizing Frequency Assignment in a Changing TDMA/FDMA Cellular Environment

Dong-Wan Tcha, June-Hyuk Kwon, Taek-Jin Choi, and Se-Hyun Oh

***Abstract*—A robust operation of assigning frequencies to requirements in a time-division multiple-access (TDMA) [or frequency-division multiple-access (FDMA)] cellular system should cope with environmental changes such as short-term demand rises and long-term capacity expansions while always keeping the required realignment process as simple as possible. In this paper, we consider the so-called *perturbation-minimizing frequency assignment problem* (PMFAP), the objective of which is to assign available frequencies for newly generated requirements with the minimum change in the existing frequency assignments while meeting the interference-related constraints. For PMFAP, we propose a heuristic algorithm based on the so-called Bv–Dw rearrangement technique, which can also be applied to the classic *frequency assignment problem* (FAP) with a slight modification. Two kinds of computational experiments, one for each of the above two problems, are performed to demonstrate the powerful features of the proposed solution method not only in its suitability for real-world frequency management, but also in solving the FAP.**

***Index Terms*—Assignment with rearrangement, frequency assignment problem, TDMA/FDMA.**

I. Introduction

The recent demand explosion for mobile communication services, together with the finite frequency spectrum allocated to this service, makes the issue of frequency assignment ever more important for the design and operation of such systems. The cellular system is founded on frequency reuse between cells sufficiently apart from each other, so that it must prohibit use of the same and/or adjacent frequency channels (or simply frequencies) in each cell and its neighboring cells to prevent radio interference. Three kinds of such interference-related constraints—cochannel, adjacent channel, and cosite—have to be satisfied for the cellular system (see [1] and [2]), which are aggregately called here the compatibility constraints. The objective of frequency assignment problem (FAP) is to assign frequencies for given channel requirements with minimum frequency span while meeting the compatibility constraints. Though the FAP is originally for frequency-division multiple-access (FDMA) systems, it is also well defined and very critical for a time-division multiple-access (TDMA) system with multiple carriers (see [3]).

Manuscript received May 8, 1997; revised November 2, 1998.

D.-W. Tcha is with the Graduate School of Management, Korea Advanced Institute of Science and Technology, Seoul 130-012, Korea.

J.-H. Kwon and T.-J. Choi are with the Radio Network Engineering Team, LG TeleCom, Seoul 153-010, Korea.

S.-H. Oh is with the Digital Cellular Business Division, SK Telecom, Seoul 100-095, Korea.

Publisher Item Identifier S 0018-9545(00)02579-2.

However, the FAP is not quite suited for being directly used by the system operator which has to meet real fluctuations and temporal increases in demand with a given number of available frequencies. Consider the case when the final solution of the FAP is so different from the preassigned pattern of the operative system as to require a full-scale frequency reshuffle. The reshuffling would heavily burden the system both in time and in cost. From the system operator's point of view, the smaller the change in the preassignment, the better, particularly for a short-term fluctuation in the traffic environment. Another point hampering its direct use arises from doubting that available frequencies beyond the frequency span obtained from solving the FAP are of no value. This realization motivates us to consider the so-called perturbation-minimizing frequency assignment problem (PMFAP), the objective of which is to meet newly generated channel requirements with a given number of available frequencies while keeping the number of frequency reassignments as low as possible.

The problem of assigning frequencies for cells with newly generated requirements was studied by Lee *et al.* [4]. They considered, however, the model with only cochannel interference constraints. Furthermore, their model does not allow frequencies assigned to the existing requirements to be altered, which certainly reduces the flexibility required to accommodate more channel requirements.

In Section II, we briefly review the classic model of FAP, along with two representative solution strategies. Taking the notational framework, we then present our PMFAP formulation. In Section III, a heuristic algorithm is proposed, each iteration of which consists of three operations, requirement selection (RS), unforced assignment (UA), and forced assignment with rearrangement (FAR). The proposed algorithm for PMFAP is shown to effectively solve the FAP itself by always attempting to assign to a requirement under consideration in the algorithmic process the lowest numbered frequency in the given set of ordered available frequencies. The version for FAP is summarized in Section IV. In Section V, we report two kinds of computational experiments. The first experiment is to show the effectiveness of our algorithm for PMFAP in minimizing the number of frequency reassignments. The focus of the second one is placed on demonstrating the competitiveness of the version of our algorithm modified for FAP in solving the FAP itself. Finally, Section VI provides concluding remarks.

II. Notation and Problem Statement

Frequencies are assumed to be evenly spaced, so they can be identified with positive integers 1, 2, 3, etc. We denote the universal set of requirements by $X = \{x_1, x_2, \cdots, x_n\}$, to each

0018–9545/00$10.00 © 2000 IEEE

member of which a frequency is assigned such that the frequencies assigned to two requirements, for example, $x_i, x_j \in X$, should be separated by at least c_{ij}. Traditionally, the whole requirements are grouped into cells, based on which the compatibility constraints are defined with indexes referring to cells. However, for exposition brevity, we let the compatibility constraints be represented by compatibility matrix $C = \{c_{ij}\}$ as defined above with indexes referring to requirements. Then integer c_{ij} denotes the cosite constraint when two requirements x_i and x_j belong to the same cell, whereas both adjacent and cochannel constraints correspond to the case where two requirements belong to different cells.

Given set X and $n \times n$ matrix C, a frequency assignment is represented by n vector $F = (f_i)$, where component f_i represents the frequency assigned to requirement $x_i \in X$. For future use, a two-tuple (x_i, f_i) will be called a requirement-frequency pair (or a r-f pair) and the set $\{(x_i, f_i)|x_i \in X\}$ a *requirement-frequency match* (RFM). A frequency assignment F is called a *feasible (frequency) assignment*, or an RFM is *feasible*, if it satisfies the following condition:

$$|f_i - f_j| \geq c_{ij}, \qquad \text{for all } x_i, x_j \in X \text{ and } i \neq j.$$

Then, the objective of FAP is to find a feasible assignment F such that

$$S(F) := \max_{x_i \in X} f_i \quad \text{(i.e., the required frequency span)}$$

is as small as possible.

The FAP is NP-complete, since it can be viewed as a generalization of the NP-complete graph coloring problem [1], [6]. The traditional solution approach for FAP has thus been on developing heuristics by extending the graph coloring algorithms [2], [5], [7]–[9]. Most of such FAP heuristics are based on sequential assignment of frequencies according to some heuristic order. That is, the typical strategy is to rank whole requirements in descending order of assignment difficulty and then to assign frequencies one by one to the ordered requirements via a number of differing assignment rules. The paper of Sivarajan *et al.* [5] may be the representative of such studies. They defined the degree of a cell and used it as a measure on the difficulty of assigning frequencies to requirements in the cell. They further proposed four requirement ordering methods based upon this measure and used two representative assignment strategies which are frequency exhaustive strategy (FES) and requirement exhaustive strategy (RES).

For PMFAP, we denote the set of available frequencies as $L = \{1, 2, \cdots, l\}$. Let X^0 with $m = |X^0|$ be the set of initially given requirements for each component x_i of which f_i^0 has been assigned. Given such an RFM$^0 = \{(x_i, f_i^0)|x_i \in X^0\}$, we aim at finding a feasible assignment F or a feasible RFM for a set X expanded from X^0.

For a feasible assignment F of X, we define, for every $x_i \in X^0$,

$$I(x_i) := \begin{cases} 1, & \text{if } f_i \neq f_i^0, \\ 0, & \text{otherwise.} \end{cases}$$

Then, the objective of PMFAP is to find a feasible frequency assignment $F = (f_i)$ or a feasible RFM for X such that

$$P(F) := \sum_{x_i \in X^0} I(x_i)$$
$$\text{(i.e., the level of perturbation on the initial assignment)}$$

is as small as possible.

Throughout this study, we shall employ the requirement-frequency matching rule to assign the lowest numbered frequency whenever possible. Under this rule, it would be instructive to distinguish between two cases, one with a sufficiently large number of available frequencies and the other not. In the former case, an algorithm for PMFAP would proceed to minimize the associated frequency span, which also makes itself fit for solving an FAP. Consider the latter case where there are not enough available frequencies to meet the compatibility constraints of the requirement set X, as often encountered in practice. The natural goal of this type of PMFAP would be to minimize the number of requirements which remain unmatched and then to reduce the level of the above perturbation. This preemptive preference of two objectives is well answered by the algorithmic process that all the available frequencies are always orderly and exhaustively searched to obtain a feasible r-f pair for each requirement.

III. Heuristic Algorithm for PMFAP

In this section, we propose a heuristic algorithm for PMFAP. Each iteration of our algorithm consists of three operations: requirement selection (RS), unforced assignment (UA), and forced assignment with rearrangement (FAR). Let $Q(\subseteq X)$ be the set of requirements to each member of which a certain frequency is already assigned during the algorithmic process, including the preassigned requirement set X^0, and let $\overline{Q} = X \backslash Q$. A requirement-frequency pair, (x_i, f_i) for $x_i \in \overline{Q}$, is said to be *feasible* to the present RFM if it does not conflict with the RFM in terms of the compatibility constraints, i.e.,

$$|f_i - f_j| \geq c_{ij}, \qquad \text{for all } (x_j, f_j) \in \text{RFM}.$$

For the purpose of explaining the three operations, it will be convenient to introduce the following definition.

Definition 1: Consider a subset of the RFM, the constituent r-f pairs of which conflict with the r-f pair (x_i, f_i), $x_i \in \overline{Q}$. We shall call the subset of requirements in Q, corresponding to the pair, as the f_i blocker of x_i, denoted as $B(x_i, f_i)$. So if the frequency assignments for requirement set $B(x_i, f_i)$ are undone, then f_i becomes a feasible frequency for x_i, or the r-f pair (x_i, f_i) becomes feasible, with the remaining pairs of the RFM.

If $B(x_i, f_i) = \emptyset$ for $f_i \in L$, then f_i is immediately assignable to x_i, symbolized as $f_i \rightarrow x_i$. The case of nonempty $B(x_i, f_i)$ is denoted by $f_i \tilde{\rightarrow} x_i$.

The RS operation selects a requirement which is most difficult to assign a frequency among the requirements in $\overline{Q}$, and this difficulty measure is a generalization of the saturation degree proposed by Brélaz [10] for GCP. For this, the assignment

difficulty of an unassigned requirement x_i ($\in \overline{Q}$) is defined as the number of infeasible frequencies with respect to the RFM, that is, the number of f_i's such that $f_i \not\rightarrow x_i$.

Once an unassigned requirement x_i is selected, the UA operation finds the lowest frequency in the list L feasible to the RFM. That is, if an f_i ($\in L$) such that $B(x_i, f_i) = \emptyset$ is found, a successful assignment is made to proceed to the next iteration. Otherwise, the FAR operation takes over.

It is the FAR operation that makes the proposed algorithm prominent. Assume that for an $x_i \in \overline{Q}$, there is no f_i with $f_i \rightarrow x_i$ with respect to the present RFM. Then, the FAR attempts to assign a frequency in L to requirement x_i with the minimum enforcing overhead, i.e., with the minimum perturbation on the present RFM. The essence of FAR is to identify a subset $R(x_i)$ of Q, which is defined as follows.

Definition 2: The $R(x_i)$ for x_i ($\in \overline{Q}$) is defined as a minimal subset of Q, each requirement of which can be simultaneously reassigned with an alternative feasible frequency, so that the x_i can be assigned a frequency feasible to the realigned RFM.

There can be a number of $R(x_i)$'s for an x_i ($\in \overline{Q}$) from its definition as a minimal subset of Q. To identify one such $R(x_i)$, we examine a sequence of f_i blockers for $f_i \in L$ until a termination criterion is met. Each time a $B(x_i, f_i)$ of x_i is generated, we undo the corresponding portion of frequency assignment in the RFM and try to assign an alternate feasible frequency to each requirement of $B(x_i, f_i)$ by the UA operation. If these frequency reassignments are successfully made, $B(x_i, f_i)$ becomes $R(x_i)$ by itself.

In case such a frequency reassignment cannot be made for some requirement x_j in $B(x_i, f_i)$, one proceeds to identify $B(x_j, f_j)$ and attempts to reassign an alternate feasible frequency to each $x_k \in B(x_j, f_j)$. Such $B(x_j, f_j)$'s are blockers at the second depth level or simply 2-DL. Generalizing this, we define the kth depth-level (k-DL) blockers for the r-f pair (x_i, f_i), $x_i \in \overline{Q}$.

Now suppose that k-DL blockers have been successively identified up to level l to make way for assigning f_i to x_i ($\in \overline{Q}$). Let $B(x_p, f_p)$ be a nonempty l-DL blocker. The task is then to check whether each requirement of the blocker is reassignable with an alternate feasible frequency. For the systematic checking of ($l + 1$)-DL blockers, we define the *requirement-frequency fixing*, RFF$((x_i, f_i), (x_p, f_p))$, to be the set of r-f pairs, which have been successively matched along the vertical path from (x_i, f_i) down to (x_p, f_p) in the RFF search tree. Thus, when at the $(l + 1)$-DL, we identify those blockers of r-f pairs which do not conflict with those pairs in RFF$((x_i, f_i), (x_p, f_p))$. Note that the cardinality of RFF$((x_i, f_i), (x_p, f_p))$ is not less than its path length, as seen in the following example.

Now with the notation defined above, the FAR operation is illustrated via the following simple example in Fig. 1.

In this example, the RFM is $\{(a, 3), (b, 2), (c, 1), (d, 2), (e, 3), (f, 1)\}$ and the list of available frequencies L is $\{1, 2, 3\}$. The existence of cochannel interference between a pair of requirements is represented by an edge connecting the corresponding two nodes on the figure.

We shall first elaborate on the vertical (depth-first) search of our frequency assignment process for requirement g.

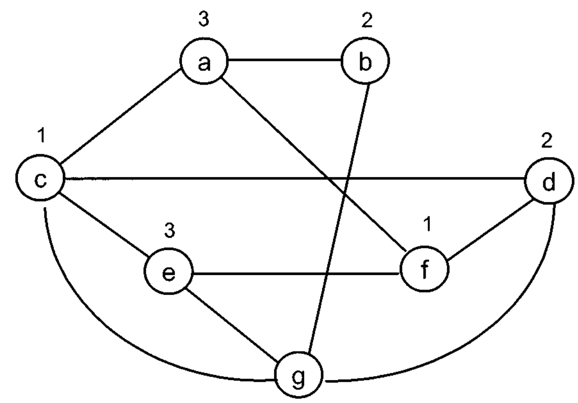


Fig. 1. Example for the FAR operation.

To assign a feasible frequency to g, we first generate three blockers: $B(g, 1) = \{c\}$, $B(g, 2) = \{b, d\}$, and $B(g, 3) = \{e\}$. Focusing on requirement c in the first blocker above, we move on to the 2-DL with the updated RFM:= $\{(a, 3), (b, 2), (d, 2), (e, 3), (f, 1), (g, 1)\}$. Since $(c, 1)$ conflicts with RFF$((g, 1), (g, 1)) := \{(g, 1)\}$, we consider only two blockers: $B(c, 2)$ and $B(c, 3)$. From $B(c, 2) = \{d\}$ and $3 \rightarrow d$, the associated vertical search is fathomed, giving rise to an $R(g) = \{c, d\}$. From $B(c, 3) = \{a, e\}$ and $2 \rightarrow e$, we move on to the 3-DL for an unassigned requirement, a, by setting RFF$((g, 1), (c, 3)) := \{(g, 1), (c, 3), (e, 2)\}$. Note that the inclusion of $(e, 2)$ in the RFF is to simplify the search, which in effect implies the nonexhaustiveness of our search. For requirement a, we have two pairs, $(a, 1)$, and $(a, 2)$, not conflicting with the RFF. Generation of 3-DL blockers shows $B(a, 1) = \{f\}$ and $B(a, 2) = \{b\}$. Searches along this vertical direction continue in this manner.

Going back to the original requirement g, recall that we have $B(g, 2) = \{b, d\}$ and $B(g, 3) = \{e\}$. The UA operation shows $1 \rightarrow b$, $3 \rightarrow d$, and $2 \rightarrow e$. For these two r-f pairs, $(g, 2)$ and $(g, 3)$, two sets of $R(g)$'s are found as $\{b, d\}$ and $\{e\}$, respectively. Note that we have another set of $R(g)$, $\{c, d\}$, obtained as above, shown in Fig. 2.

From the above simple example, we have witnessed how often the process of checking the reassignability of an f_i blocker, is repeatedly executed in generating an $R(g)$. The following lemma shows the complexity of this process.

Lemma 1: The problem of checking the reassignability of a nonempty blocker $B(x_i, f_i)$ is NP-complete.

Proof: Suppose that the frequency assignments for $B(x_i, f_i)$ are undone, rendering the RFM reduced that much. For each requirement $x_j \in B(x_i, f_i)$, let $L(x_j)$ ($\subseteq L$) be the set of frequencies feasible to the reduced RFM. Without loss of generality, let the compatibility constraint for any two requirements in $B(x_i, f_i)$ be either zero or one. The checking problem under this restriction becomes a GCP on graph $G = (V, E)$, where node v_j corresponds to requirement x_j in $B(x_i, f_i)$, edge (v_j, v_k) is defined if $c_{jk} = 1$, and colors of $L(x_j)$ are available for node v_j. The GCP is known to be NP-complete, and so is the checking problem. □

The FAR operation is encoded to render the so-called vth breadth-level and wth depth-level (Bv–Dw) procedures. In the

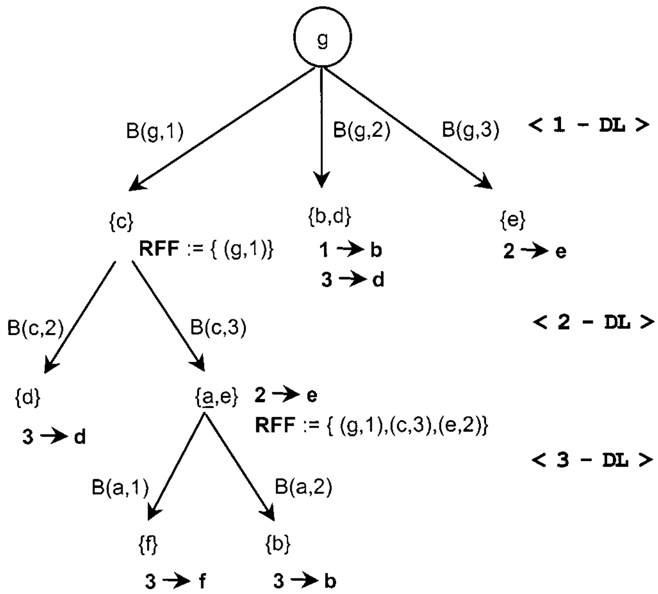


Fig. 2. RFF search tree of the example.

Bv–Dw procedure, we consider blockers only within the cardinality of v (i.e., $|B(x_i, f_i)| \leq v$) and limit the number of successive downward searches to w. This can be visualized via the associated RFF search tree: each node can have at most v dangling nodes and the path lengths are at most w.

The reassignabilities of blockers at the same depth level are systematically checked in the increasing order of blocker cardinality. The procedure starts from $B1$, moves on to $B2$, and continues in this manner until Bv at the $D1$ level and then repeats this process for each unassigned requirement in blockers of the next lower level. However, the complexity of FAR operation actually prohibits the direct implementation of this general Bv–Dw procedure. Viable alternative search strategies would be accentuating either of two tracks: breadth and depth. Two procedures, Br–$D1$ and $B1$–Dr, then arise.

Procedure: Br–$D1$ procedure of the FAR operation.

Step 0) Start with the given RFM^0 and an unassigned requirement $x_i \in \overline{Q}$.

Step 1) Find $\{B(x_i, f_i) | \forall f_i \in L\}$ and set $k := 1$.

Step 2) If $k > r$, terminate (/* fail to produce a feasible assignment */). Otherwise, let

$$L_k := \{f_i \in L || B(x_i, f_i)| = k\}.$$

Step 3) If $L_k = \emptyset$, set $k := k + 1$ and go to Step 2). Otherwise, select a frequency f^* in L_k.

Step 4) If all requirements in $B(x_i, f^*)$ are found reassignable, update the RFM as found and stop. Otherwise, set $L_k := L_k \backslash \{f^*\}$ and go to Step 3).

Considering the NP-completeness of the reassignability check, it is certainly not worthwhile to continue the search beyond a certain breadth level, say Br, though there is a slim chance of the reassignability of the remaining large blockers. At this point, it is instrumental to note that the search pattern not only saves the associated computational burden, but also conforms with our objective of minimizing the level of perturbation on the present RFM.

For the other procedure, $B1$–Dr, identifying blockers and checking their reassignability get more complex as the depth level increases. So instead of listing the $B1$–Dr procedure, we elaborate on the $B1$–$D2$ procedure which limits the search within the $D2$ level.

Procedure: $B1$–$D2$ procedure of the FAR operation.

Step 0) Start with the given RFM^0 and an unassigned requirement $x_i \in \overline{Q}$.

Step 1) ($D1$ level) Call the $B1$–$D1$ procedure. If x_i is assigned, update RFM and stop. Otherwise, let

$$L^1 := \{f_i \in L || B(x_i, f_i)| = 1\}.$$

Step 2) If $L^1 = \emptyset$, terminate (/* fail to produce a feasible assignment */). Otherwise, select the frequency f^* in L^1 and set RFF:= $\{(x_i, f^*)\}$.

Step 3) ($D2$ level) For $x_j \in B(x_i, f^*)$, let

$$L^2 := \{f_j \in L || B(x_j, f_j)| = 1 \\ \& B(x_j, f_j) \text{ not conflicting with RFF}\}.$$

Step 4) If $L^2 = \emptyset$, set $L^1 := L^1 \backslash \{f^*\}$ and RFF:= $\emptyset$, and go to Step 2). Otherwise, select the frequency f' in L^2.

Step 5) If the requirement in $B(x_j, f')$ is found reassignable, update the RFM as found and stop. Otherwise, set $L^2 := L^2 \backslash \{f'\}$ and go to Step 4).

IV. New Algorithm for FAP

The FAR operation can be directly employed to solve the classic FAP, in which RFM^0 $:= \emptyset$ according to our setting. The proposed algorithm will be called the frequency exhaustive strategy with rearrangement (FESR), which are obtained by modifying the FES [2], [5].

Recall that the FES for FAP proceeds as follows: starting at the top of the ordered requirement list, it consecutively assigns the lowest possible frequency which does not violate the compatibility constraints with the assignments already made. Let S_k denote the current frequency span after completing the assignment operations of step k in the FES process. At every step k, if the lowest feasible frequency f^* is less than or equal to the current span S_{k-1}, the FES can then assign a frequency without increasing the current span; otherwise, the increase of the span is inevitable in the FES. The former operation is called the frequency-insert and the latter the frequency-append.

This FES is a straightforward implementation of the sequential graph coloring algorithm [12]. The weakness of such an algorithmic process is the rigidity that r-f pairs, once matched, cannot be reshuffled later on. The FESR modifies the rigid FES by substituting the FAR operation for each frequency-append. A simple, but effective FESR algorithm is now presented, where the employed FAR operation is the $B1$–$D1$ procedure. Listing of a general FESR algorithm incorporating a Bv–Dw procedure is also avoided for exposition brevity.

Algorithm: Frequency exhaustive strategy with rearrangement.

Step 0) Start with the ordered requirement list. Let $k := 1$, $S_k := 1$, and RFM $:= \emptyset$.

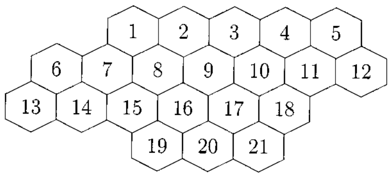


Fig. 3. The 21-cell system.

Step 1) Set $k := k+1$. If $k \leq n$, find the lowest frequency f^* for x_k feasible to the RFM. Otherwise, stop (/* a solution is at hand */).

Step 2) (*Frequency-Insert*) If $f^* \leq S_{k-1}$, update RFM by adding r-f pair (x_k, f^*). Set $S_k := S_{k-1}$ and go to Step 1).

Step 3) (*Frequency-Rearrangement*)

a) Let $f := 1$.

b) If $f < f^*$, obtain a set $B(x_k, f)$. Otherwise, go to Step 4).

c) If $|B(x_k, f)| = 1$, find the lowest frequency $\hat{f}$ for $\hat{x} \in B(x_k, f)$ that $(\hat{x}, \hat{f})$ is feasible to the reduced RFM and (x_k, f). Otherwise, set $f := f+1$ and go to Step b).

d) If $\hat{f} < f^*$, update RFM by adding both (x_k, f) and $(\hat{x}, \hat{f})$. Set $S_k := \max\{S_{k-1}, f, \hat{f}\}$ and go to Step 1). Otherwise, $f := f+1$ and go to Step b).

Step 4) (*Frequency-Append*) Update RFM by adding (x_k, f^*) and $S_k := f^*$, and go to Step 1).

Note that our term for the *frequency-insert* is the UA operation in PMFAP, while our FAR is associated with the *frequency-rearrangement*.

V. Computational Results

The 21-cell system in [5] and [11], the configuration of which is given in Fig. 3, is still used to generate test problems. To demonstrate the powerful features of the proposed algorithm, we perform two kinds of computational experiments. The first experiment aims at showing the effectiveness of our algorithm for PMFAP in minimizing the level of perturbation, i.e., of requirement-frequency rematches for RFM^0. Its effectiveness, however, may not be sufficiently illustrated due to the nonexistence of a study in the literature directly dealing with the PMFAP. Motivated by this realization, we perform another experiment, the focus of which is placed on demonstrating the superiority of FESR, the version of our algorithm modified for FAP, over the existing FAP algorithms in solving the FAP itself.

A. Experiment on PMFAP

We consider two scenarios on channel requirements as shown in Table I. The first one typifies the situation of short-term demand rises, whereas the second one a system expansion with newly added cells. The first column of each scenario shows the number of existing requirements to which frequencies are already assigned, and the second one the increased number of aggregate requirements.

As for the interference constraints in this experiment, we use, to facilitate the comparisons with the existing studies, the traditional cell-based values for the adjacent channel and cosite constraints which are, respectively, set at two and five for simplicity. For the cochannel constraint characterized by N_c, however, we consider two values of N_c, 12 and 7, to check the effect of the severity of interference environment. Note the differences of parameter values of constraints from those of our requirement-based formulation of PMFAP. For example, $N_c = 12$ here implies that the same frequency can be reused between two cells with radius distance of six. The number of available frequencies l is also varied between two values to check the effect of frequency availability.

TABLE I
CHANNEL REQUIREMENTS FOR THE 21-CELL SYSTEM

Cell	Number of Requirements			
	Scenario 1		Scenario 2	
	existing	new	existing	new
1	10	12	11	14
2	12	15	9	13
3	10	14	13	13
4	11	11	10	12
5	12	13	0	9
6	12	14	0	7
7	13	16	8	11
8	11	12	9	10
9	12	17	12	14
10	10	11	10	11
11	13	15	11	13
12	14	15	0	8
13	13	16	0	9
14	15	19	14	15
15	12	13	8	10
16	12	12	12	13
17	13	17	10	11
18	10	15	11	12
19	10	12	0	6
20	15	18	13	13
21	15	17	0	8
total	255	304	161	232

With this input data specification, Tables II and III summarize the performance of our algorithm. Throughout this experiment, we set the initial requirement-frequency match, denoted by RFM^0, as the best solution with the minimum span among those generated by the FAP algorithms listed in [5]. To demonstrate the superiority of solutions generated by our algorithm, two measures are used: the number of requirements which remain unmatched even after running our algorithm, denoted by $U(X \backslash X^0)$, and the number of matches altered from the initial RFM^0, denoted by $P(\text{RFM}^0)$. Table II shows an interesting phenomenon: the quality of solutions generated by the algorithm with $B2$–$D1$ is not noticeably better in both measures than the most basic algorithm with $B1$–$D1$, whereas the $B1$–$D2$ outperforms the $B1$–$D1$ though with extra computational burden. The plain performance of $B2$–$D1$ after the application of $B1$–$D1$ may be reasoned as follows: the RFM obtained after the $B1$–$D1$ rearrangement is so densely ordered that reshuffling the existing matches with a large cardinality may sometimes do harm on the existing compact match pattern.

Table III is provided to additionally accentuate how well our algorithm serves its objective of minimizing the level of perturbation. For the new aggregate demand of each scenario, we

TABLE II
COMPARISONS BETWEEN VARIOUS VERSIONS OF OUR ALGORITHM

Scenario	$N_c^{1)}$	$l^{2)}$	Algorithm with B1-D1			Algorithm with B2-D1			Algorithm with B1-D2		
			$U(X \setminus X^0)^{3)}$	$P(RFM^0)^{4)}$	CPU$^{5)}$	$U(X \setminus X^0)$	$P(RFM^0)$	CPU	$U(X \setminus X^0)$	$P(RFM^0)$	CPU
1	7	130	19	7	2.5	19	11	3.1	16	20	40.1
		150	0	2	0.3	0	2	0.4	0	2	0.3
	12	160	22	11	4.7	23	14	5.1	15	42	60.6
		180	2	4	1.1	2	4	1.1	0	12	5.3
2	7	100	14	5	1.4	15	7	1.5	12	11	10.8
		120	1	1	0.3	0	3	0.3	0	3	0.6
	12	120	32	5	3.3	32	5	3.6	32	13	82.2
		140	4	4	0.8	4	4	0.9	3	5	18.7

1) cluster size
2) number of available frequencies
3) number of requirements which remain unmatched
4) number of matches altered from RFM^0
5) computation time in seconds

TABLE III
COMPARISONS BETWEEN THE EXISTING FAP AND OUR ALGORITHMS

Scenario	N_c	Sivarajan et al.		Algorithm with B2-D1		Algorithm with B1-D2	
		span*	$P(RFM^0)$	$U(X \setminus X^0)$	$P(RFM^0)$	$U(X \setminus X^0)$	$P(RFM^0)$
1	7	150	227	0	2	0	2
	12	181	227	1	4	0	6
2	7	124	139	0	1	0	1
	12	144	135	0	7	0	7

* the minimum among the spans obtained by eight algorithms of Sivarajan et al.

TABLE IV
PERFORMANCE COMPARISONS ON FAP

Problem	LB	Sivarajan et al. [5]	Wang et al. [15]	Sung et al. [16]	FESR with B1-D1	FESR with B1-D2
1	533	533	533	533	533	533
2	533	533	533	533	533	533
3	381	381	381	381	381	381
4	427	445	433	436	434	433
5	529	529	529	529	529	529
6	309	310	309	309	309	309
7	258	270	263	268	265	260

applied the well-known algorithm for FAP by Sivarajan *et al.* [5] to obtain the RFM. The first column shows the associated span, and the second the relatively large value of $P(\mathrm{RFM}^0)$ indicating the full-scale reshuffling on the existing match. Taking this span as our l, the results of the algorithm with $B2$–$D1$ and $B1$–$D2$ are listed in Table III. It is remarkable to find the outstanding records of two algorithms in both performance measures.

B. Experiment on FAP

Seven FAP examples widely used in the literature are again adopted to compare the performance of our FESR on solving the FAP with those of the well-known FAP algorithms [5], [15], [16]. A version of the FESR is made for each of four requirement ordering methods in [5]. Each performance record listed in Table IV is the best one among those obtained by the above four versions of the corresponding algorithm. The column of LB shows the tightest lower bound among those in [11], [13], and [14].

The figures of resulting frequency spans in Table IV show that our FESR, particularly that with $B1$–$D2$, outperforms all the existing FAP algorithms, even the most recent one with significant margin in some cases. This is indeed an impressive performance record of the FESR on solving the FAP when considering that the FESR is simply a modification of the algorithm for solving the PMFAP.

VI. CONCLUDING REMARKS

We addressed the so-called PMFAP, the objective of which is to meet newly generated channel requirements with a given number of available frequencies while keeping the level of perturbation as low as possible. The PMFAP was defined to suit the real-world frequency assignment practices in TDMA and FDMA cellular systems where the level of perturbation is preferred to be kept minimal for any demand fluctuation.

For PMFAP, we have developed a heuristic algorithm based on the Bv–Dw type of frequency rearrangement technique. The proposed algorithm, though originally developed for PMFAP, can be applied to the FAP with a slight modification.

To demonstrate the powerful features of the proposed solution methods, we perform two kinds of computational experiments. The first experiment is to show the effectiveness of our algorithm on solving the PMFAP, and the second one the competitiveness of the modified version on solving the classic FAP. This computational experience convinces us that the algorithms based on the Bv–Dw search not only are efficient enough to be directly used for real-world frequency management by system operators, but also outperform all the existing FAP algorithms even on solving the classic FAP itself.

REFERENCES

[1] W. K. Hale, "Frequency assignment: Theory and applications," *Proc. IEEE*, vol. 68, pp. 1497–1514, Dec. 1980.

[2] J. A. Zoellner and C. L. Beall, "A breakthrough in spectrum conserving frequency assignment technology," *IEEE Trans. Electromag. Compat.*, vol. EMC-19, pp. 313–319, Aug. 1977.

[3] K. Raith and J. Uddenfeldt, "Capacity on digital TDMA systems," *IEEE Trans. Veh. Technol.*, vol. 40, 1991.

[4] H. Lee, S. B. Kim, and S. I. Lee, "Additional channel assignment in a cellular system by graph theoretic model," in *Proc. ITC*, 1994, pp. 471–480.

[5] K. N. Sivarajan, R. J. McEliece, and J. W. Ketchum, "Channel assignment in cellular radio," in *Proc. 39th IEEE Veh. Technol. Soc. Conf.*, May 1989, pp. 846–850.

[6] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York: Freeman, 1979.

[7] F. Box, "A heuristic technique for assigning frequencies to mobile radio nets," *IEEE Trans. Veh. Technol.*, vol. VT-27, pp. 57–64, 1978.

[8] A. Gamst and W. Rave, "On frequency assignment in mobile automatic telephone systems," in *Proc. GLOBECOM '82*, pp. 309–315.

[9] W. K. Hale, "New spectrum management tools," in *IEEE Proc. Int. Symp. EMC*, Aug. 1981, pp. 47–53.

[10] D. Brélaz, "New methods to color the vertices of a graph," *Commun. ACM*, vol. 22, pp. 251–256, Apr. 1979.

[11] A. Gamst, "Some lower bounds for a class of frequency assignment problems," *IEEE Trans. Veh. Technol.*, vol. 35, pp. 8–14, Feb. 1986.

[12] D. W. Matula, G. Marble, and J. D. Isaacson, "Graph coloring algorithms," in *Graph Theory and Computing*, R. C. Read, Ed. New York: Academic, 1972, pp. 109–122.

[13] C. W. Sung and W. S. Wong, "A graph theoretic approach to the channel assignment problem in cellular systems," in *Proc. 45th IEEE Veh. Technol. Soc. Conf.*, 1995, pp. 604–608.

[14] D. W. Tcha, Y. J. Chung, and T. J. Choi, "A new lower bound for the frequency assignment problem," *IEEE/ACM Trans. Networking*, vol. 4, Dec. 1996.

[15] W. Wang and C. K. Rushforth, "An adaptive local-search algorithm for the channel-assignment problem (CAP)," *IEEE Trans. Veh. Technol.*, vol. 45, pp. 459–466, Aug. 1996.

[16] C. W. Sung and W. S. Wong, "Sequential packing algorithm for channel assignment under cochannel and adjacent channel interference constraint," *IEEE Trans. Veh. Technol.*, vol. 46, pp. 676–686, Aug. 1997.

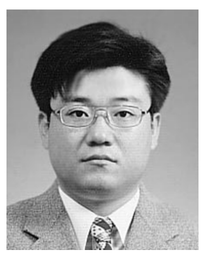
Dong-Wan Tcha received the B.S. degree in electronics from Seoul National University, Seoul, Korea, in 1969 and the M.S. and Ph.D. degrees in operations research from Northwestern University, Evanston, IL, in 1972 and 1975, respectively.

In 1975, he joined the faculty of the Korea Advanced Institute of Science and Technology, Seoul, where he is currently a Professor at the Graduate School of Management. He was a Visiting Scientist at the IBM T. J. Watson Research Center from 1981 to 1982, a Guest Scientist at the German Aerospace Research Establishment (DFVLR) in 1986, and a Humboldt Fellow at the Technical University of Darmstadt, Germany, in 1987. His research has been in the areas of network analysis, combinatorial optimization, and communication systems. He has published numerous papers on operations research in the IEEE TRANSACTIONS ON COMMUNICATIONS, the IEEE TRANSACTIONS ON COMPUTERS, the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, and the IEEE TRANSACTIONS ON NETWORKING.

Dr. Tcha served as the Editor of the *Journal of Korea Operations Research and Management Science Society* from 1983 to 1985 and has been the President of that Society since 1998. He also served as a Director responsible for academic activities at the Korean Institute of Communication Sciences and as the Chairman of the SIG on Telecommunications Management from 1993 to 1994. Currently, he is a Member of the Advisory Editorial Board of the *Journal of Information Technology and Management*. He will serve as a General Cochair at the INFORMS 2000 Seoul International Conference.

June-Hyuk Kwon received the B.S. degree in business administration from Yonsei University, Seoul, Korea, in 1991 and the M.S. and Ph.D. degrees in Management Science from the Korea Advanced Institute of Science and Technology, Taejon, Korea, in 1993 and 1998, respectively.

Currently, he is with LG Telecom, Seoul, where he is involved with the wireless network design of CDMA cellular systems.



Taek-Jin Choi received the B.S. degree in economics from Yonsei University, Seoul, Korea, in 1988 and the M.S. and Ph.D. degrees in management science from the Korea Advanced Institute of Science and Technology, Taejon, Korea, in 1991 and 1996, respectively.

In 1996, he joined LG Telecom, Seoul, where he has been in charge of the team for wireless network design and optimization.

Se-Hyun Oh received the B.S. degree in 1982 from Seoul National University, Seoul, Korea, and the M.S. and Ph.D. degrees from the Korea Advanced Institute of Science and Technology, Taejon, Korea, in 1984 and 1992, respectively.

Currently, he is a Senior Manager and Distinguished Research Engineer at the New Technology Development Division of SK Telecom, Seoul. He joined SK Telecom in 1992 and has led the development of design and operation technology of AMPS and the IS-95 cdmaOne system. He currently works on strategic planning and development of IMT-2000 systems and services. His current research interests are in radio transmission technology and protocols and hardware and software design of wireless communication systems.